Volume: 03; Issue: 03; Pages:1169-1175 (2011)

Development of 8-lane PCI-Express protocol using VHDL

K. Harish, H.S Aravinda
Department of Electronics and Communication Engineering
REVA Institute of Technology and Management,
Kattigenahalli, Yelahanka, Bangalore, Karnataka, India- 560064
Harik03@yahoo.com, aravindhs@yahoo.com

----------**ABSTRACT**----------
Flosolver Mk 8 is the latest family member of the Flosolver series of parallel computers in CSIR-NAL that is currently being developed, to have a performance of 10 TFLOPS with 1024 processors in it. It is based on distributed memory concept, using quad core xeon processors[11]. Each cluster consists of 8 processors, a FPGA based Floswitch, and 4 PCI cards. The inter Cluster communication is carried out through optical transceivers to provide high speed communication. PCI is used for interface between the server and the FloSwitch. Unlike any other switch, the Floswitch has the capability of performing information processing operation which is a unique feature, along with message passing[12]. To this existing system the project intends to replace the PCI card with 8-lane PCI-Express add-on card. The PCI-Express defines a line rate of 2.5Gbps per lane.

The basic goal of this project entitled "Development of PCI-Express protocol using VHDL" is to Design and Develop a PCI-Express protocol for a 8x PCI-e card, with an optical transceiver and DPM (Dual Port Memory) as an external interfaces. The development includes the generation of 8 x PCI-e cores and interfacing the core for optical transaction and also for the DPM transaction. The PCI-Express add-on card contains a FPGA (Virtex V– XC5VLX110T) and the card supports 8X lane. FPGA provides an interface between the PCI-Express signals, the DPM and the optical transceiver module. The protocol has to be developed using VHDL and simulated using model sim 6.1f.

Keywords – **PCI-Express, FPGA,DPM, Optical transceiver**

Paper submitted: July 08,2011 Date of Acceptance: August 13,2011

1. INTRODUCTION

PCI Express is the third generation high performance I/O bus used to interconnect peripheral devices in applications such as computing and communication platforms. The first generation buses include the ISA, EISA, VESA, and Micro Channel buses, while the second generation buses include PCI, AGP, and PCI-X. PCI Express is an all encompassing I/O device interconnect bus that has applications in the mobile, desktop, workstation, server, embedded computing and communication platforms[5].

1.1. PCI Express Link

A Link represents a dual-simplex communications channel between two components. The fundamental PCI Express Link consists of two, low-voltage, differentially driven signal pairs: a Transmit pair and a Receive pair as shown in Figure 1-1[5].

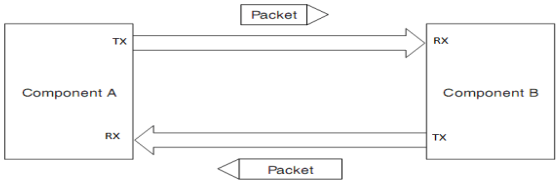


Figure 1.1: PCI express link

A PCI Express interconnect that connects two devices together is referred to as a Link. A Link consists of either x1, x2, x4, x8, x12, x16 or x32 signal pairs in each direction. These signals are referred to as Lanes. A designer determines how many Lanes to implement based on the targeted performance benchmark required on a given Link.

1.2. PCI Express Layering Overview

The PCI-express specification defines a layered architecture in terms of three discrete logical layers: the Transaction Layer, the Data Link Layer, and the Physical Layer. Each of these layers is divided into two sections: one that processes outbound (to be transmitted)

information and one that processes inbound (received) information, as shown in Figure 1-2-1[2].

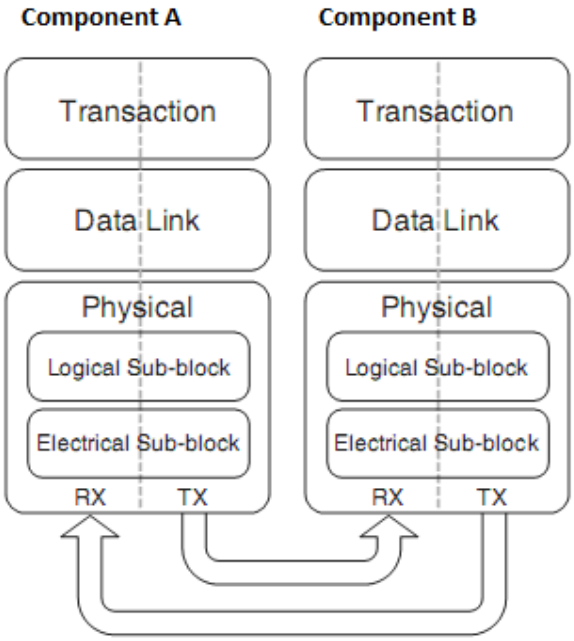
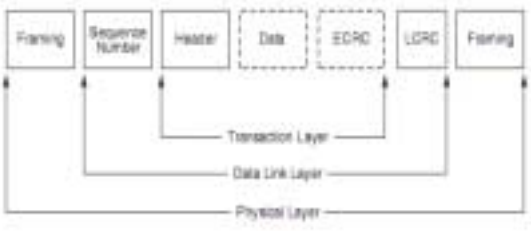


Figure 1.2.1: PCI EXPRESS LAYERING.

PCI Express uses packets to communicate information between components. Packets are formed in the Transaction and Data Link Layers to carry the information from the transmitting component to the receiving component. As the transmitted packets flow through the other layers, they are extended with additional information necessary to handle packets at those layers. At the receiving side the reverse process occurs and packets get transformed from their Physical Layer representation to the Data Link Layer representation and finally (for Transaction Layer Packets) to the form that can be processed by the Transaction Layer of the receiving device[5].



• Figure 1.2.2: Packet flow through layers

1.2.1 Transaction Layer

The upper Layer of the architecture is the Transaction Layer. The Transaction Layer's primary responsibility is the assembly and disassembly of Transaction Layer Packets (TLPs). TLPs are used to communicate transactions, such as read and write, as well as certain types of events . The Transaction Layer is also responsible for managing credit-based flow control for TLPs[1].

Every request packet requiring a response packet is implemented as a split transaction. Each packet has a unique identifier that enables response packets to be directed to the correct originator. The packet format supports different forms of addressing depending on the type of the transaction (Memory, I/O, Configuration, and Message). The Packets may also have attributes such as No Snoop and Relaxed Ordering.

The transaction Layer supports four address spaces: it includes the three PCI address spaces (memory, I/O, and configuration) and adds a Message Space. This specification uses Message Space to support all prior sideband signals, such as interrupts, power-management requests, and so on, as in-band Message transactions. You could think of PCI Express Message transactions as "virtual wires" since their effect is to eliminate the wide array of sideband signals currently used in a platform implementation.

1.2.2 Data Link Layer

The middle Layer in the stack, the Data Link Layer, serves as an intermediate stage between the Transaction Layer and the Physical Layer. The primary responsibilities of the Data Link Layer include Link management and data integrity, including error detection and error correction.

The transmission side of the Data Link Layer accepts TLPs assembled by the Transaction Layer, calculates and applies a data protection code and TLP sequence number, and submits them to Physical Layer for transmission across the Link. The receiving Data Link Layer is responsible for checking the integrity of received TLPs and for submitting them to the Transaction Layer for further processing. On detection of TLP error(s), this Layer is responsible for requesting retransmission of TLPs until information is correctly received, or the Link is determined to have failed.

The Data Link Layer also generates and consumes packets that are used for Link management functions. To differentiate these packets from those used by the Transaction Layer (TLP), the term Data Link Layer Packet (DLLP) will be used when referring to packets that are generated and consumed at the data link layer.

1.2.3 Physical Layer

In this node, we have a temperature sensor, a PIR sensor, a The Physical Layer includes all circuitry for interface operation, including driver and input buffers, parallel-to-serial and serial-to-parallel conversion, PLL(s), and impedance matching circuitry. It includes also logical functions related to interface initialization and maintenance. The Physical Layer exchanges information with the Data Link Layer in an implementation-specific format. This Layer is responsible for converting information received from the Data Link Layer into an

appropriate serialized format and transmitting it across the PCI Express Link at a frequency and width compatible with the device connected to the other side of the Link.

1.2.4 PCI-EXPRESS ADD-ON CARD

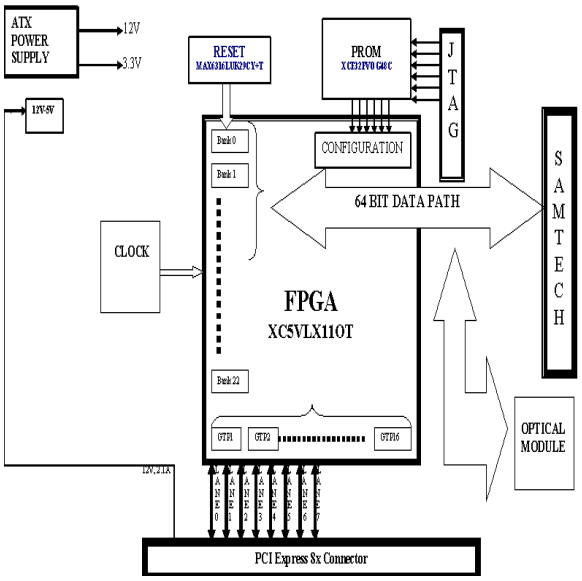


Figure 1.2.4: PCI-EXPRESS add on card

Roll of PCI-express add-on card in super computer Mk-VIII

1.PCI-express add-on card provide the speed communication link between PEs and communication Floswitch.

2.This IPC card is to interface the PCI-express signals to DPM and optical interface.

3.The IPC card should fit as an add-on card in the 8X PCI-express slot of the Intel's Server motherboard

4.The card provides required connections for testing and debugging.

5.To do necessary address decoding, interfacing of control signals and data lines.

6.It is a plug and play compatible.

.7Acting as a communication device by reading and writing data to/from DPM.

2 LITERATURE SURVEY

2.1 Study of Existing System

Flosolver Mk-8 is the latest family member of the Flosolver series of parallel computers that is currently being developed in CSIR-NAL to have a peak performance of 10 TFLOPS with 1024 processors in it based on Intel SR1530CLR boards, based on Distributed memory concept and built around Intel Quad Core Xeon Processors, which acts as processing elements [PEs]. The communication between processing elements is very important in parallel processing. In the existing system communication between PE's is achieved through the Floswitch. The interfacing between PE's and the switch is done through the 64 bit PCI-card[3].

To this existing system, the proposed project aims at replacing the current PCI add-in card with the PCI-Express 8X add-in card[4].

2.2 BRIEF REVIEW OF LITERATURE

PCI-Express BUS

PCI Express is the third generation high performance I/O bus used to interconnect peripheral devices in applications such as computing and communication platforms. PCI Express is an all encompassing I/O device interconnects bus that has applications in the mobile, desktop, workstation, server, embedded computing and communication platforms. PCI Express interconnect architecture is primarily specified for PC based (desktop/laptop) systems. But just as PCI, PCI Express is also quickly moving into other system types, such as embedded systems.

A single PCI Express link is a dual-simplex connection using two pairs of wires to transmit only one bit per cycle. Although this sounds limiting, it can transmit at extremely high speed of 2.5 Gbps, which equates to 320 Mbps on a single connection. These two pair of wires is called a Lane. This allows significant reduction in the pin count while maintaining or improving throughput. It also reduces the size of the PCB, the number of traces and layers, and simplifies layout and design.

PCI Express uses a serial interface and allows for point-to-point interconnections between devices using directly wired interfaces between these connection points. This differs from the previous PCI bus architectures that used shared, parallel bus architecture.

DPM (Dual Port Memory)

The DPM is a static RAM having two ports A and B with access time of 12/15 ns. In this Project, the IDT70V658 is a high-speed 64K x 36 Asynchronous Dual-Port Static RAM. The IDT70V658 is designed to be used as a stand-alone 2304K-bit Dual-Port RAM or as a combination MASTER/SLAVE Dual- Port RAM for 72-bit-or-more word system. Using the IDT MASTER/ SLAVE Dual-Port RAM of 72-bit or wider memory system application will result in full-speed, error-free operation without the need for additional discreet logic.

This device provides two independent ports with separate control, address, and I/O pins that permit independent and asynchronous access for reads or writes to any location in memory. An automatic power down feature controlled by the chip enables (either CE0 or CE1) permit the on-chip circuitry of each port to enter a very low standby power mode. The 70V658 can support an operating voltage of either 3.3V or 2.5V on one or both ports, controlled by the OPT pins. The power supply for the core of the device (VDD) remains at 3.3V.

2.3 SCOPE OF STUDY

The scope of this project is to develop PCI-Express protocol and interface the 8 X PCI-Express add-on cards to the DPM and to the optical interface.

2.4 METHODOLOGY

The methodology that has been followed for the design is:

- Design of state machine
- Develop the VHDL code for design.
- VHDL simulated output waveforms will be tested for verification.

3 KEYWORK

3.1 PCI-EXPRESS TRANSACTIONS

3.1.1TRANSMISSION FROM PCI-EXPRESS TO OPTICAL MODULE

When PCI-express wants to transfer the data to the optical module, it first initializes all the internal pointers and the signals necessary to do this job. Once this is done, the data from the PCI-express is continuously read and written to the FIFO, and transmitted on to the optical channel.
The state machine for transmission of data from the PCI-e to the optical module consists of the following states & the flow chart is represented in the figure 3.1.1

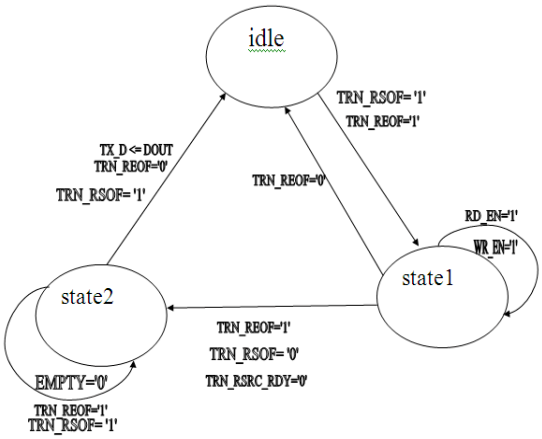


Figure 3.1.1 : FLOW CHART FOR TRANSMISSION TO OPTICAL MODULE

IDLE :- It is a global reset state. During this state all the internal pointers and output and input registers are initialized and also all the signals pertaining to PCI-express core and the aurora core is deasserted.

STATE1:-In this state, the data is received from the PCI-e core and is transmitted to the optical module through the FIFO and hence asserts the signal trn_rsrc_rdy indicating the source is ready to transmit the data and also asserts the signal trn_rsof indicating the start of the frame. This state also checks whether there is a single data to be transmitted or multiple data's. The necessary signals are asserted and de-asserted accordingly. In this state the data is written and read from the FIFO onto the data lines TX_D of the aurora and enters the nextstate.

STATE2:-This state keeps track of the last data available from the core by monitoring the trn_reof signal, which indicates the transmission of the last data and until this signal is asserted the data is been transmitted to the optical link through the FIFO.

3.1.2 TRANSMISSION FROM OPTICAL MODULE TO PCI-EXPRESS

When optical module needs to transfer the data to the PCI-Express, it first initializes all the internal pointers and the signals necessary to do this job. Once this is done, the data from the optical interface is continuously written to the FIFO and transmitted on to the PCI-Express data lines.

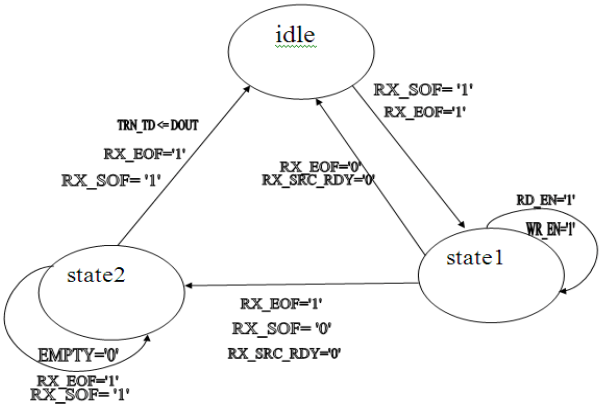


Figure 3.1.2: FLOW CHART FOR TRANSMISSION TO PCI-e FROM OPTICAL MODULE

IDLE :- It is a global reset state. During this state all the internal pointers and output and input registers are initialized and also all the signals pertaining to PCI-express core and the aurora core is deasserted.

STATE1:- In this state the aurora core accepts data from the optical link and transmits data packets to the PCI-e core through the FIFO and hence asserts the signal RX_SRC_RDY_N indicating the source is ready to transmit the data and also asserts the signal RX_SOF_N indicating the start of the frame. This state also checks whether there is a single data to be transmitted or multiple

data's. The necessary signals are asserted and de-asserted accordingly. In this state the data is written and read from the FIFO onto the data lines TRN_TD of the PCI-e core and enters the next state.

STATE2:- This state keeps track of the last data available from the aurora by monitoring the RX_EOF_N signal which indicates the transmission of the last data and until then keeps on writing data to the FIFO lines and the necessary signals are asserted, and transmits the data on the data lines TRN_TD of the PCI-e core through the FIFO. Once the last data is written to the PCI-e core the state machine is returned back to the IDLE state.

3.1.3 PCI-EXPRESS READS DATA FROM DPM

When PCI-express wants to read the data from the DPM, it first initializes all the internal pointers and the signals necessary to do this job. Once this is done, the data from the DPM is continuously read and written to the FIFO, and transmitted to the PCI-Express data lines.

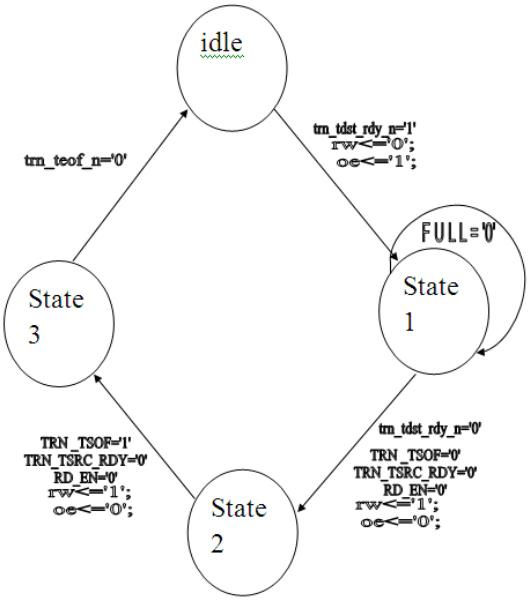


Figure 3.1.3: FLOW CHART FOR TRANSMISSION FROM DPM

IDLE: - It is a global reset state. During this state all the internal pointers and output and input registers are initialized. The base address of the DPM is initialized in this state.

STATE1:- In this state the data from the DPM is continuously read and transmits data to the PCI-e core through the FIFO. The signal which does the job of reading from the DPM and writing it to the FIFO is asserted respectively in this state. In this state the PCI-e core is ready to accept data from the FIFO and hence asserts the signal trn_tsrc_rdy indicating the source is ready to accept the data and also asserts the signal trn_tsof indicating the start of the frame and enter next state.

STATE2:- In this state the data from the DPM is continuously read and transmits data to the PCI-e core through the FIFO. In this state signal trn_tsof is deassertted.

STATE3:- This state keeps track of the last data available from the DPM by monitoring the TRN_TEOF_N signal which indicates the transmission of the last data and until then keeps on writing data to the FIFO lines and the necessary signals are asserted, and transmits the data on the data lines TRN_TD of the PCI-e core through the FIFO. Once the last data is written to the PCI-e core the state machine is returned back to the IDLE state.

3.1.4 PCI-EXPRESS WRITES THE DATA TO DPM

When PCI-express wants to write the data to the DPM, it first initializes all the internal pointers and the signals necessary to do this job. Once this is done, the data from the PCI-express is continuously read and written to the built-in FIFO, and written to the DPM.

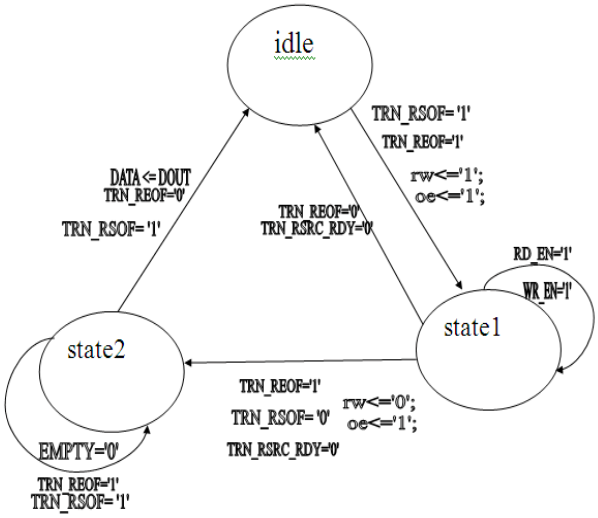


Figure 3.1.4: FLOW CHART FOR WRITING TO DPM

IDLE :- It is a global reset state. During this state all the internal pointers and output and input registers are initialized and also all the signals pertaining to PCI-express core and the DPM is deasserted.

STATE1:-In this state, the data is received from the PCI-e core and is transmitted to the DPM through the FIFO and hence asserts the signal trn_rsrc_rdy indicating the source is ready to transmit the data and also asserts the signal

trn_rsof indicating the start of the frame. This state also checks whether there is a single data to be transmitted or multiple data's. The necessary signals are asserted and de-asserted accordingly. In this state the data is written and read from the FIFO onto the data lines DATA of the DPM and enters the nextstate.

STATE2:-This state keeps track of the last data available from the core by monitoring the trn_reof signal, which indicates the transmission of the last data and until this signal is asserted the data is written to the DPM through the FIFO.

3.2 TESTING AND VERIFICATION

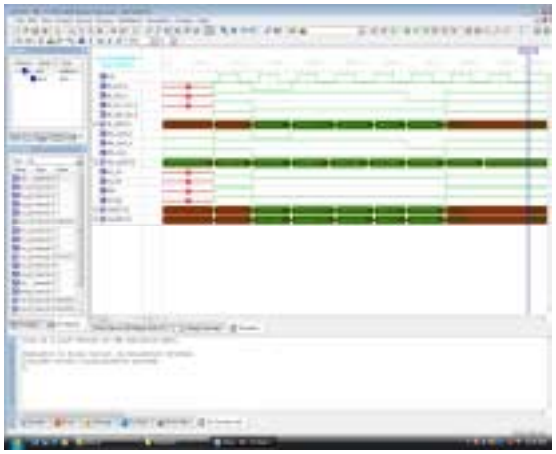


Figure 3.2.1: Receiving data from PCI-Express and writing to the optical data lines

The figure 3.2.1 shows the different signals used when the PCI-Express wants to transmit the data to the single channel optical module present on the PCI-Express add on card.

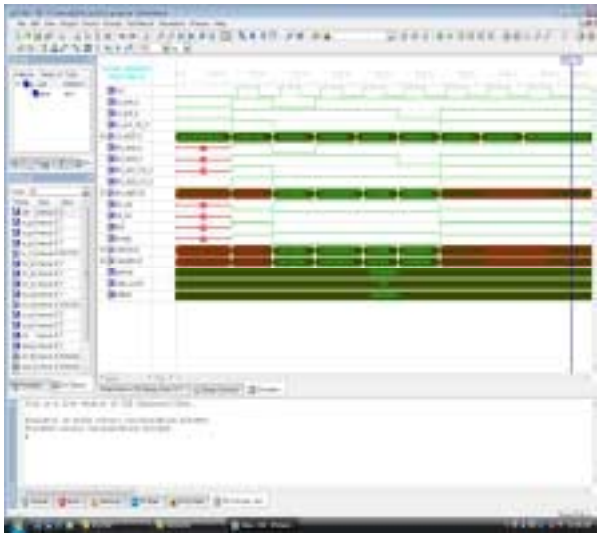


Figure 3.2.2: Receiving data from optical module and writing to PCI-express lines

The figure 3.2.2 through shows the different signals used when the single channel optical module wants to transmit the data to the PCI-Express data lines.

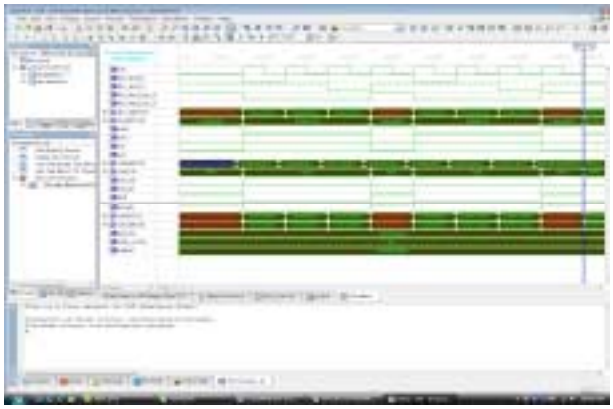


Figure 3.2.3: PCI EXPRESS reading data from the DPM

The figure 3.2.3 shows the different signals used when the PCI-Express wants to read the data from the DPM (Dual Port RAM).

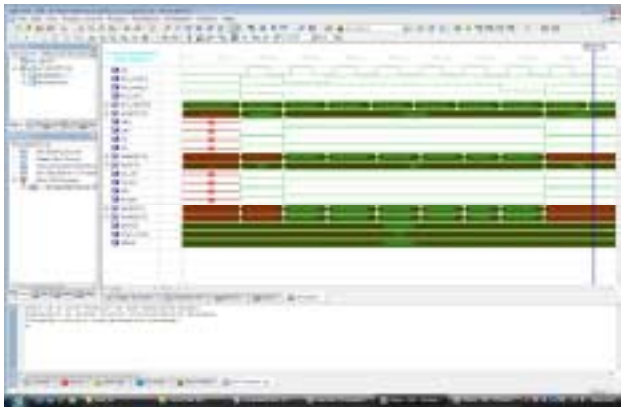


Figure 3.2.4: PCI EXPRESS writing data to the DPM

The figures 3.2.4 shows the different signals used when the PCI-Express wants to WRITE the data to the DPM (Dual Port RAM).

4 CONCLUSION

The project entitled **"**DEVELOPMENT OF 8-LANE PCI-EXPRESS PROTOCOL USING VHDL **"**, is undertaken to Design and Develop a PCI-Express protocol for a 8x PCI-e card, with a optical transceiver and DPM (Dual Port Memory) has a external interfaces. The development includes the generation of 8 x PCI-e cores and interfacing the core for optical transaction and also for the DPM transaction.

Volume: 03; Issue: 03; Pages:1169-1175 (2011)

5 FUTURE ENHANCEMENT

The PCI-Express card has presently been designed and developed for 8x, and the protocol for this has been developed, this can be further extended to 16x and 32x based on our requirements which will drastically improve the throughput, and also the protocol can be enhanced for more number of external optical link.

In the future, PCI Express communication frequencies are expected to double and quadruple to 5 Gbits/sec and 10 Gbits/sec.

ACKNOWLEDGEMENTS

We would like to express our sincere thanks to Asst professor **Arvinda.H S** , ECE department, Reva ITM and **Jagannadham V V** Scientist, Flosolver Unit CSIR-NAL, Bangalore for his continued support and guidance towards the concept. His continuous feedback has always been the strongest motivation behind this work.

REFERENCES

[1] P. Germann, M. Doyle, R. Ericson, S. Lewis, J. Dangler, A. Patel," Pushing the Limits of PCI-Express:A PCIe Application within an IBM Supercomputing Environment", IEEE transaction 2008.

[2] David Mayhew and Venkata Krishnan, "PCI Express and Advanced Switching: Evolutionary Path to Building Next Generation Interconnects", IEEE transaction 2003.

[3] Faya Peng, "INTEGRATING PCI EXPRESS INTO PXI AND VXI FOR HIGH PERFORMANCE SYSTEMS", IEEE transaction 2007.

[4] Jagan, Anand, Shashi, Rekha, Vinodini, Rajesh, "VERILOG codes for 64bit PCI Card" PDFS 0703-MAR-2007

[5] Tom Shanley and Don Anderson, PCI System Architecture, Mindshare, Addison-Wesley publishing company, 3rd edition 1995

[6] Xilinx,Inc.http://www.xilinx.com/products/virtex5/index.htm

[7] Peter Ashenden - VHDL Design and Synthesis, third edition

[8] Chang.K.C. – Digital Design and Modeling with VHDL and Synthesis, first edition

[9] J.Bhaskar – VHDL Primer, third edition.

[10] VHDL code for Virtex-5 FPGA based OpticalBoard. NalAL-PDFS0823.Anandaraj.D, Sundara Rao.G, Rekha Nila, Gautham Keshri.R and Senthil Nathan.J.

[11] U.N.Sinha, Deshpande M.D., Sarasamma V.R. "Flosolver, A Parallel Computer for fluid dynamics", Volume 57 Pages 1271-1285, 1988.

[12] U.N.Sinha, Deshpande M.D., Sarasamma V.R. – "Flosolver, A Parallel Computer", Supercomputer. Volume 4, Pages 37-42, July 1989

Authors Biography

K.Harish received B.E. degree in Electronics and Communication Engineering from the Visvesvaraya Technological University, Belgaum and pursuing M.Tech degree from Visvesvaraya Technological University, in the field of VLSI Design and Embedded Systems. His research interests are VLSI based Agent applications.

H S Aravinda received B.E. degree in Electronics and Communication Engineering from the Visvesvaraya Technological University, Belgaum and completed M.Tech degree from Mysore University, Mysore in the field of Bio-Medical Instrumentation. He is having 14 years of experience in the teaching field. He Submitted the Ph D thesis in the field of fault tolerance to VTU, Belgaum and waiting for the defense. He has published 15 papers in different journals and conferences. His research interests are Fault tolerance, Signal processing, communication.